

DBPLUS Performance Monitor description of changes in version 2018.1.1



Table of Contents

1	New	features in version 2018.1.1	3
	1.1 F	ormatting vertical axis labels on charts	.3
	1.2 T	he capability to change the database in the Database Analysis module	.3
	1.3 C	hange in Sgl Analyze	.4
	1.4 Ir	nprovement in Memory tab	.4
	1.5 S	ession view	.5
	1.6 H	listory screen of active sessions and UNDO using sessions	.5
	1.7 lr	nprovements in the SQL Find option	.6
	1.7.1	Displaying additional statistics for searched queries	. 6
	1.7.2	Improvements in query analyses, which change the execution plan	. 7
	1.8 lr	nprovement in Show Plan Objects	.8
	1.9 U	pdating Application architecture screen	.8
	1.10 C	hanges in DBPLUSCATCHER monitoring service	.9
	1.10.1	Problem with monitoring of query statistics	9
	1.10.2	2 Implementation of a restructuring process for tables in Dashboard screen	10
	1.10.3	3 Statistics monitored at day level	10
	1.10.4	Deleting history from a table storing lock history	10
	1.10.5	5 Slow operation of the use assessment process for UNDO, taking into account	
	REUS	SE blocks	10
	1.11 C	hanges in the configurator	10
	1.11.1	Adding a database to monitoring and permission verification	10
	1.11.2	2 User name verification	11
	1.12 G	seneral optimisation	11
	1.12.1	Changing size/height of charts	11
	1.12.2	2 Detecting problems with non-functioning monitoring	12
	1.12.3	3 Improper sorting in Slow SQL screen	12
	1.12.4	Group Literals/Ungroup literals filter in Database Load, Slow Select screens	12
	1.12.5	Sqi Details – displaying with an option without grouping by period and filter by hour 12	S
	1.12.6	5 Latches screen – lack of calculated waiting times per second	13
	1.12.3	Backups screen	13



Presented below are changes in DBPLUS Performance Monitor system for monitoring Oracle databases.

1 New features in version 2018.1.1

1.1 Formatting vertical axis labels on charts

In previous versions of the system, large figures were abbreviated – in some situations the shortened numbers could misrepresent the actual values, e.g. 3M value on Y axis for Disk Reads [MB] statistic.





The following chart presents the new version of the system:



1.2 The capability to change the database in the Database Analysis module

Upon entering the Performance module of the system through *Database Analysis* on the upper bar of the screen, identifiers of a selected database show up.

A functionality was added here, which allows switching between monitored databases.

DB+ Better performance	REPOSITORY													
Sack to dashboard	Sessions Sort usage sessions Undo usage sessions Sessions history Session / Undo history													
Performance	 Active sessions 	Active sessions 🗹 Users only Min elapsed time: 0 📥 sec. Sid: Username:												
E I/O Stats							Show a	dditional filter	5					
Space monitor	SELECT SESSION (LAST REFRESHED: 11:36:30) Kill session													
Memory	Logon time	Sid Serial	Hash Value	Username	Status	Elapsed	Schema	OS user	Process	Process	Machine	Pr		

Clicking on the "database" icon results in the display of a table with monitored databases:

DBPLUS

C	B+ Better performance	REPO	SITORY	latabase on MAQC	H server 💽					
-	Back to dashboard SELECT AND CHANGE WORKING DATABASE Search database by name									
	Performance	mance								
	- Chomanoo	Host	Instance	Database SID	Used database name	Туре				
	I/O Stats	name	name				Show	w add	ditional filter	s
	Space monitor	MAQCH	xe	XE	REPOSITORY	Not Specified				
	Space monitor	MAQCH	xe	XE	XE	Not Specified	homo		05 upor	D
	Memory						nema		03 080	(se
£	Sessions							PLI I	JSŁUGA L	112
	Sessions	-								
	 Session Resources 									

After switching databases, the system remains in the same screen. The option improves system ergonomics, and, in particular, allows rapid assessment of certain indicators/parameters in selected databases.

1.3 Change in Sql Analyze

An additional % indicator displaying the share of selected queries in relation to load was added to the Sql Analyze module. The information is available after hovering the cursor over the chart area.



1.4 Improvement in Memory tab

An additional summary indicator was added to charts displaying occupied and free memory space.

This summary is visible, among others, in the SGA tab, after clicking on the Shared Pool size field.



Below is an example from the SGA History tab for an SGA MAX chart.





1.5 Session view

In RAC type databases in all tabs of the Sessions screen, namely:

- Sessions
- Sort usage session
- Undo usage session
- Session History

the results and filters now have a field with instance identifier (previously only visible in the first tab).

Below is an example of the Undo usage session tab:

	Sessions	Sort us	age se	ssions	Undo usage se	essions	Sessions history	Sessio	on / Undo histo	ry					
	Show segm	ent info (si	lower)										All inst	ances 👻	Refresh
U	NDO USAGE SESSIONS (LAST REFRESHED: 12:53:53) Kill session														
								Und	o segments						≡
	8 100 9 50								50						
	Size														
	0								Undo size						
	Used Free														
Inst Id	Logon 1	ime	Sid	Serial	Hash Value	Username	Status	Elapsed Time [Seconds]	Schema	OS user	Machine	Program	Wait	Used records	Used blocks
1	2018-06-04	09:23:54	129	1589	1497696986	DBPLUS	INACTIVE	4.0	DBPLUS	radoslaw	MAQCH	SQL Dev	SQL*Net message f	6 993	120

Below is an example of the Session / Undo history tab:

Sessions	Sort usage sessions	Undo usage sessions	Sessions history	Session / Undo history					
From: 🔟 2	2018/06/04 00:00 to:	2018/06/04 23:59	Using Hash Value/Sql	Id: Enter hash value or sql i	d Username:	Enter username	Sid:	Refresh	^
				Hide additional filters					
All instanc	es 👻	Machine:			Module:				

Adding a filter allows displaying the session for a designated database instance.

Additionally, **upon the first entry** to Sessions screen there was a problem with displaying the session for the wrong instance – only after clicking the Refresh button the system displayed sessions for the current database – this inconvenience was fixed.

1.6 History screen of active sessions and UNDO using sessions

In the History screen of active sessions and sessions using UNDO space, cumulative indicators for monitored sessions were added:



In the main screen, the following information was added:

- Number of sessions using UNDO space
- Number of records/changes in UNDO
- Size of the changes in UNDO

Logdate -	Active Sessions	Undo Usage Sessions	Undo Used Record Count	Undo Used Blocks Count [Blocks]
2018/06/04 13:29:50	116	2101	121333	3 848
2018/06/04 13:28:44	138	2117	138342	4 303
2018/06/04 13:27:37	124	2147	120759	3 881
2018/06/04 13:26:30	131	2120	127985	3 930
2018/06/04 13:25:20	124	2099	111907	3 701
2018/06/04 13:24:14	132	2120	135776	4 378
2018/06/04 13:23:08	107	2094	110909	3 694
2018/06/04 13:22:01	137	2099	120954	3 828

1.7 Improvements in the SQL Find option

The query search module, SQL Find, can be found in SQL Details screen.

Instance Load	Waits	Latches	SQL Analyze	SQL Details	Load Trends	Compare Trends	Top SQL	SQL 3D	Top Day	Slow SQLs	Perf Counters
Enter query hash	Fro	om: 📰 201	8/04/23 00:00	to: 📰 2018/04/	23 23:59	Group by plan hash	Group by [Day 👻 🗆	Online values	Refresh	Find SQL
STATEMENT TEXT											

The following improvements were added to this option:

1.7.1 Displaying additional statistics for searched queries

The following information was added to every tab:

- Query duration time / Elapsed Time
- Virtual processor use time / Cpu Time
- Number of executions
- Number of read blocks:
 - from drive devices
 - o from memory
- Number of returned records

Presented below is the result of an example search by query text:

Statement by text		select%transac	tion						×					
Plan Flip-Flop State	ments	D-14 6	0040/04/02				May	raturnad etatamante:	100					
New statements		Date Irom.	2010/04/23	00.00		.016/04/25	23.59	inter statements.	100 +					
CLICK ON [ADD TO SQL	LICK ON [ADD TO SQL DETAILS] BUTTON (ICON WITH +) TO ADD QUERY IDENTIFIER TO QUERY HASHES TOOLBAR LIST													
Query Hash	Last execution date	Elapsed Time [\$econds]	Cpu Time [\$econds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Buffer writes [Blocks]	Rows processed						
0x165730E7D548B0AB	2018-04-23	24.77	20.85	394	0 MB	455 858	0	18 518	elect * from dbo.ProductWithTransactio					
0x603C0FBB24BFFE94	2018-04-23	2 637.67	2 261.19	46 500	9 MB	53 800 500	0 46 5		SELECT COUNT(*) FROM [dbo].[Produ					
0x64C102F23329DC98	2018-04-23	14 818.37	8 970.26	86 530	124 MB	4 916 333 563	0 1 155 521 620		elect top(@v) t.* from Production.Trans					
0x677E3020F458809A	2018-04-23	6 816.98	5 671.19	445 436	8 MB	391 092 808	0	445 436	elect top ? * from Production.Transactic					
0xC234CB9C6A04DF5D	2018-04-23	4.19	3.73	3 915	0 MB	0	0	3 915	elect @tc = COUNT(*) FROM sys.dm_t					
0xCBE0580B79F27DC6	2018-04-23	276.83	240.54	19 613	0 MB	3 443 399	9 0 156 90		9 0 156 90		9 0 156 96		elect p.ProductNumber, p.Name, p.Mał	
0xD54F433E1DEA7406	2018-04-23	1.16	1.15	40 280	0 MB	161 120	0	40 280	SELECT COUNT(*) FROM [Production].					
0xD9C52915FFDD7EF2	2018-04-23	18.54	15.76	3 945	0 MB	8 594	0	33	elect SessionTrans.session_id, s.status					
0xDB32CA35859FA837	2018-04-23	91.28	73.76	7 138	0 MB	1 253 174	0	57 104	elect p.ProductNumber, p.StandardCos					
•														



Presented below is the result of an example search of queries executed on a certain day but not executed on the previous day:

Statement by text		Statement exe	cuted in period					х					
Plan Flip-Flop State	ements	Date from	2018/04/2	3 00:00	Date to	2018/04/23	23:59	Min. elapsed time (sec): 100 *					
New statements		And statement not executed in the period range											
		Date from:	2018/04/2	2 00:00	Date to:	2018/04/22	23:59	Search					
CLICK ON [ADD TO SQL DETAILS] BUTTON (ICON WITH +) TO ADD QUERY IDENTIFIER TO QUERY HASHES TOOLBAR LIST													
Query Hash	Elapsed Time	e Cpu Time	Executions	Disk reads	Buffer gets	Buffer writes	Rows processed	Query text					
0x64C102F23329DC98	15 181.5	9 9 178.58	88 532	124 MB	5 030 077 193		1 182 256 328	elect top(@v) t.* from Production.TransactionHistory t where t.Produ					
0x677E3020F458809A	6 816.9	8 5 671.19	445 436	8 MB	391 092 808	0	445 436	elect top ? * from Production.TransactionHistory t where t.Transactio					
0x603C0FBB24BFFE94	2 686.6	8 2 303.32	47 311	9 MB	54 738 827	0	47 311	SELECT COUNT(*) FROM [dbo].[ProductWithTransaction] WHERE [
0x2F5EE731FCEDF74A	293.7	3 228.99	859	233 MB	25 665 202	0	25 770	elect h.CustomerID, h.SalesOrderID, h.OrderDate, h.ShipDate, h.Sa					
0x4BC9BB522ADAD20F	120.1	0 118.87	47 311	1 MB	6 055 808	0	47 311	elect count(*) from Sales.vIndividualCustomer where BusinessEntity					
4	_			_		_	_	• • • • • • • • • • • • • • • • • • •					

1.7.2 Improvements in query analyses, which change the execution plan

For queries changing the execution plan, additional information grouped in accordance with the areas below were added:

- Statistics with a summary for all execution plans where the query was active
- Statistics with a summary for the slowest plan
- Statistics with a summary for the fastest plan
- Comparison of the slowest and the fastest
- Estimation of a possible reduction in the query duration time

Presented below is an example with a search result of those queries, which changed the execution plan within the period of two weeks:

View of areas: Total statistics,	Slowest	plan	statistics
----------------------------------	---------	------	------------

Statement by text						404				×			
Plan Flip-Flop Stat	ieme	Date from:	2018/04/10	Date to:	2018/0	4/24 23:59							
New statements										Search			
CLICK ON [ADD TO SQL DETAILS] BUTTON (ICON WITH +) TO ADD QUERY IDENTIFIER TO QUERY HASHES TOOLBAR LIST													
			Total statis	tics			Slov	vest plan statistics					
Query Hash	Query text	Elapsed Time	Cpu Time	Executions	Number of plans	Plan Hash	Elapsed Time	Cpu Time	Executions	Elapsed Time Per 1 exec			
		[Seconde]	[Seconds]				[Seconds]	[Seconds]		[Seconde]			
0x64C102F23329DC98	select top(@v)	486 925.70	394 926.91	426 821	2	0x31F605092B25	442 606.41	367 606.00	156 096	2.8355			
0xA86C6E5BE207D6E8	select max(Erro	70.20	24.68	43	2	0x397376A5E330	52.39	19.52	21	2.4946			
0x25B65C61193863C4	select * from P	11 726.99	10 221.35	1 176 774	3	0xD445611DDBA	420.77	138.82	1 773	0.2373			
0xE95D16F7F24BD1F3	SELECT DB_I	68.70	60.64	6 695	2	0x2370E781E95E	25.13	22.26	1 339	0.0188			
0x24BFF45573B477FD	select convert(i	98.46	88.74	1 343	2	0x89C31130AB10	26.72	24.42	343	0.0779			
0x89EB3EE49C2797CF	select ? as rec	16.09	15.89	20 742	2	0x43B435618BC8	7.77	7.68	6 612	0.0012			
					_								
4										•			

View of areas: Fastest plan statistics, Slowest vs. Fastest, Estimation statistics.



			Fas	test plan statistics			Slo	west vs Fastest	Estimation statistics		
ins	Elapsed Time Per 1 exec	Plan Hash	Hash Elapsed Time Cpu T		Executions	Elapsed Time Per 1 exec	Times faster	Elapsed Time Per 1 exec difference	Elapsed Time to reduce	Cpu Time to reduce	
	[Seconds]		[Seconds]	[Seconds]		[Seconds]		[Seconds]	[Seconds]	[Seconds]	
6 096	2.8355	0xF02EB8B03876	44 319.29	27 320.91	270 725	0.1637	17	2.6718	417 052.5628	351 853.1681	
21	2.4946	0xFE2C0C637B8	17.82	5.16	22	0.8098	3	1.6848	35.3808	14.5963	
1 773	0.2373	0x90B998ECB7C	8 388.18	7 310.25	1 169 949	0.0072	33	0.2301	3 289.8811	2 868.4574	
1 339	0.0188	0x43E66D931657	43.57	38.38	5 356	0.0081	2	0.0106	14.2387	12.6670	
343	0.0779	0x2B459523C160	71.73	64.33	1 000	0.0717	1	0.0062	2.1186	2.3535	
6612	0.0012	0xC99C4CF8765/	8.32	8.21	14 130	0.0006	2	0.0006	3.8777	3.8421	
4										•	

A crucial area of the **Plan Flip-Flop Statements** screen is *Statistics estimation*. **Elapsed Time to reduce** and **Cpu Time to reduce** columns are a calculation of a possible duration reduction in cases when a query would be active only at the highest execution plan.

Helpful tip:

Sorting by one of these columns will allow finding those queries whose optimisation would result in the highest performance improvement.

1.8 Improvement in Show Plan Objects

In the Show Plan Objects screen, which allows the analysis of an execution plan, an option to display object definition was added.

The option is available after selecting the Load object properties filter and after choosing the Info tab.

SQL TEXT				EXPLAIN PLAN							
MERCE /*+ use_nl(t8 t4) USING (SELECT num9,num10 num16, Sum(num17) num17, Sum(n Sum(num27) num27 FROM deplus_tab4 t4 NHEE t4.snap_id= :snap_id A GROUP BY num9,num10,serv ON (t8.dat1 = :beginDat t4.num10)	<pre>*/ INTO dbplus_tab8 t8 0,server_id, Sum(num13) nu um18) num18, Sum(num19) nu NE ID t4.server_id = :server_ rer_id) t4 AND t8.num1 = t4.num8 AN</pre>	mi3,Sum(num14) num14,Sum(m19,Sum(num20) num20 , m id D t8.server_id = t4.serve	num15) num15,Sum(num16) in(num24) num24, r_id AND t8.num12 =	* • • •	EERCE STATEMENT (Cost - 29 , Sytes - 0 , Cardinal MERCE DEPUIS_TABS WHERE DEPUIS_TABS WHESTED LOOPS (OUTER) (Cost - 29 WIEW (Cost - 2 , Sytes - 1680 WIEW (Cost - 2 , Sytes - 1680	<pre>ity - 0 , Search Columns - 0) , Bytes - 2300 , Cardinality - 10 , Search Columns - 0) , Cardinality - 10 , Search Columns - 0) - 8 , Bytes - 270 , Cardinality - 10 , Search Columns - 0) INTEX ROWID) DBDLUS_TAB4 (Cose - 7 , Bytes - 570 , Cardin)</pre>					
OBJECTS USED IN EXPLAIN PL	AN			INDEX	ES FOR SELECTED OBJECT DBPLUS.DBPLUS_TAB4						
Туре	Owner	Object Name	Alternative Object		Owner	Name					
TABLE	DBPLUS	DBPLUS_TAB4		DBPLUS	S	DBPLUS_TAB4_NUM9					
INDEX	DBPLUS	DBPLUS_TAB4_SERVER_ID		DBPLUS	s	DBPLUS_TAB4_SNAP_ID					
TABLE	DBPLUS	DBPLUS_TAB8		DBPLUS	s	DBPLUS_TAB4_SNAP_ID_NUM10					
INDEX	DBPLUS	DBPLUS_TAB8_NUM1_IDX		DBPLUS	s	DBPLUS_TAB4_SERVER_ID					
TABLE	DBPLUS_PIP	DBPLUS_TAB4	2	DBPLUS	s	DBPLUS_TAB4_SERVER_10_SNAP					
Object columns Info Properties Details for TABLE DBPLUS_DBPLUS_TAB4 To show ddl statement enter object name: DBPLUS_RC © Show © Load object properties CREATE OR REPLACE FUNCTION "DBPLUS"."OBPLUS_RC" (p_query in vARGHAR2) return varchar2 IS 1,query long; - 1,char varchar2(1); 1,in_quotes boolean default FALSE; 1,in_runter BOOLEAN default FALSE; engin Indefault FALSE; engin Indefault FALSE;											
loop l_char := substr() IF (substr(p_quer l_in_quotes :=	<pre>p_query,i,1); y,i,1) = '''' and l_in_quote FALSE;</pre>	IS) THEN				*					

1.9 Updating Application architecture screen

The screen available at the main menu level **Servers monitor -> Application architecture** was restructured and supplemented with additional information regarding process monitoring activity.

In the area to the left, some databases are available which display:

- When the last snapshot for the monitored ORACLE database was made
- The time of the most recent database activity (connection with monitoring service with ORACLE database)

The central area contains information on the current state of DBPLUSORACLECATCHER monitoring service, which provides information such as:



- Whether the service is active
- The last activity of the service
- Memory use on a machine where the monitoring service is activated
- Processor usage by monitoring service

Below the statistics, the history of the service state can be checked for a selected time period.

The area on the right contains information on the SQL instance, where the repository of DBPLUS Performance Monitor system is located.

Below is an example slide:

List of monitored databases			N	lonitoring service		Dbplu	Dbplus Performance Monitor			
Database name	Last snapshot date	Last activity								
DBMON1@XE	2018/06/04 13:43:52	2018/06/04 13:51:15		-	1					
orcl	no snapshot in last 3	no activity in last 5 mi								
REPOSITORY	2018/06/04 13:43:52	2018/06/04 13:51:15								
хе	2018/06/04 13:43:52	2018/06/04 13:51:15								
A23@A23 (Excluded from monitoring)				DBPLUS Oracle Ca	atcher					
A24@A24 (Excluded from monitoring)			L	Service status	Running	-				
A25@A25 (Excluded from monitoring)			É	Last service activity	2018-06-04 13:50:59	×	Repository Information			
				Machine Total Memory	16375 MB		Database SID: xe			
				Machine Memory Usage	10388 MB		Host: localhost			
A27@A27 (Excluded from monitoring)				DBPLUSCATCHER Memory Usage	71 MB					
A29@A29 (Excluded from monitoring)				DBPLUSCATCHER CPU Usage	0,1 %					
A30@A30 (Excluded from monitoring)				Refresh View servic	e activity					
A31@A31 (Excluded from monitoring)										
A32@A32 (Excluded from monitoring)										

The slide below displays the history of DBPLUSCATCHER service activity, after the **View service activity** button is clicked on:



1.10 Changes in DBPLUSCATCHER monitoring service

1.10.1 Problem with monitoring of query statistics

In the database version oracle 12, there was a problem with statistic monitoring for queries, which were removed from the cache and reappeared in v\$sql system view.

The oracle database engine returns statistics with identical values, i.a., for columns Elapsed Time, CPU Time (default values are zero), which resulted in invalid reporting of query load.

The problem applied to version oracle 12 and occurred in 1-5 cases in every 1000 queries.



1.10.2 Implementation of a restructuring process for tables in Dashboard screen

A restructuring mechanism was implemented for selected tables used by the monitoring service. The process is launched once a week.

1.10.3 Statistics monitored at day level

For a query in table DBPLUS_TAB2_DAY launched with MERGE clause, a problem with changing the execution plan to a worse one occurred – the optimiser of the database used NESTED LOOP operator instead of HASH JOIN – the problem has been solved in the new version.

For Load Trends statistic a new data gathering mechanism by day was implemented – this is intended to, i.a., improve the operation rate of the performance trends screen.

1.10.4 Deleting history from a table storing lock history

The process of deleting history data from a table storing data from lock snapshots was made available.

For information purposes – the length of history by snapshots is by default retained for 30 days – the parameter can be configured in Configuration->Settings.

Because the table (i.e. DBPLUS_TAB22) was not previously cleaned, and to avoid multiple data deletion operations, the upgrade process has been restructured and preserves the history from the last 3 days.

1.10.5 Slow operation of the use assessment process for UNDO, taking into account REUSE blocks

Database optimiser for the dba_undo_extents system view could use the wrong execution plan – HASH JOIN operator, which was not selected there – the problem has been solved.

1.11 Changes in the configurator

1.11.1 Adding a database to monitoring and permission verification

While adding a database to monitoring, the configurator allowed the use of an existing user, which was then used for monitoring purposes.



Include/Add oracle database Specify an oracle instance and account	to monitoring process with admin rights which lets wizard to do co	Infiguration
Ins	stance	Finish
You need to specify the oracle in You can skip this step and every	stance that would be included in the monito time you can add/remove the database to/	ring process. from monitoring process.
Connection Type TNS	•	
Set an user account with administ It will be used to perform user creatic Authentication Username	n Oracle Authentication	se existing user
Password	Test credentials	
Step 1 from 3		Back Continue

Upon verification, the permissions CONNECT and DBA_CATALOG_ROLE were tested directly on the user. In the case when a user had an added role with such permissions, the mechanism did not allow the use of such an account – the problem has been solved in the new version.

1.11.2 User name verification

While adding a database to monitoring there was a problem when a user name began with numbers, and the configurator did not allow the use of such an account – the problem has been solved.

1.12 General optimisation

1.12.1 Changing size/height of charts

In screens such as

- Instance Load
- Waits
- Latches
- Sql Analyze
- Load Trends
- Performance Counters
- IO Stats
- Space Monitor

There is now an option to change the height of the chart – an example with a scaled chart in Load Trends screen is presented below:

DBPLUS

Instance	e Load V	'aits Latc	es SQL A	nalyze SQL Deta	ils Load Trend	is Compare	Trends Top	SQL SQL 30	D Top Day	Slow SQLs	Perf Counters			
Date from: Date Show statistics for: All databases Group by Day - Refresh													sh	
SQLINSTANCE TRENDS LOAD Chart type: Area 👻														
SQL Instance trends statistics 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0														
SQL IN STA	NCE TRENDS	STATI STIC S											Clear sel	lection
Logdate 🔺	Elapsed Time	Cpu Tim	Executi	ons Disk reads	Disk reads	Buffer gets	Buffer writes	Rows	Latches	Waits	Locks	Sessions	Active session	s
	[HH:MI:\$\$]	[HH:MC:	5]	[Blocks]	[Auto Bytee]	[Blocks]	[Auto Bytee]	[Auto]	[Seconds]	[Seconds]	[HH:MI:SS]			
2018-04-10	59:49:	32 45:	2 430 3	08 220 11 247 4	12 86 GB	17 438 286 263	450 MB	4.29 B	20	48 787	01:48:30	37		3
2018-04-11	01:53:	44 01:	9:55 104 6	81 451 791 0	95 6 GB	542 849 160	10 MB	157.49 M	0	2 247	00:00:02	33		1

1.12.2 Detecting problems with non-functioning monitoring

A mechanism of problem detection in monitoring service operation was improved in the system – a very common occurrence was the lack of space in the repository database and all information regarding service operation errors were placed in the log file on a machine with DBPLUS Performance Monitor software.

In the new version, the dashboard screen now displays an alert in cases such as

- problem with insufficient space in the repository or UNDO space
- stopped monitoring service
- lack of resources on the machine with monitoring service

Additionally, current data from the activity of DBPLUSORACLECTCHER service can be displayed in screen **Servers monitor->Application architecture**.

1.12.3 Improper sorting in Slow SQL screen

An error with the default sorting in Slow SQL functionality was fixed in the system. In the new version, after the query statistic for a selected period is displayed, the table is by default ordered by Elapsed Time column (descending sorting).

1.12.4 Group Literals/Ungroup literals filter in Database Load, Slow Select screens

In the selected screens there is a filter for activating or deactivating query grouping for the presence of literals.

Sql Statements	Waits	Alerts	Statements filter: Top 20 statements by Elapsed time 👻												UnGroup literals 👻				
SNAPSHOT OF SQL STATEMENTS EXECUTED WITHIN 15 MINUTES AT 2018-06-04 12:25:53													_						
Q Search statistic by sql text, hash value or plan hash in below snapshot table																			
Query text		Hash Value	Sql Id	Plan Hash	Elapsed Time •	Cpu Time	Time per 1 exec	Sorts	Fetches	Execution	Parse Calls	Disk reads	Buffer gets	Rows processed	Module	Number of concurrent	Db Load	Cpu Load	
					[Seconds]	[Seconds]	[Seconds]	[Rowa]	[Rows]			[k Blocks]	[Blocks]	[Rows]		users	[%]	[%]	
select * from test whe	ere id = 100	23211181 +	2vdbbhh6xbg	2283087408	1 808.39	0.00	1 808.3940	0	0	0	0	0.00 k	0	0	SQL*Plus	2	97	0	
DELETE /*+ RULE */	DBPLUS_1	1829590819	g86ax99qhupt	416993056	17.36	0.44	17.3572	0	0	1	1	5.87 k	29 772	1 653	Snapper	0	1	0	

The activity of the filter was opposite than selected in the option – this has been fixed.

1.12.5 Sql Details – displaying with an option without grouping by period and filter by hours

In Sql Details screens, if the following filters were selected:

• No group by period



• Filter setting in hour range, e.g. from 08:00 to 10:00

The system ignored the hour range and displayed statistics for the full day – this problem has been fixed.

1.12.6 Latches screen – lack of calculated waiting times per second

In the Latches screen, after clicking on a snapshot on the chart on the right side, Latch type waiting times are displayed.

The chart displayed waiting times for the total time which were not expressed per 1 second. In the new version it was corrected, as it is in the Waits screen.

1.12.3 Backups screen

In the backups screen (copies done via RMAN) there was a problem with building the chart and the capability of displaying details – this problem has been solved.